Adapting the Evaluation Space to Improve Global Learning

Alan C. Schultz
Navy Center for Applied Research in Artificial Intelligence
Naval Research Laboratory
Washington, DC 20375-5000, U.S.A.
Email: schultz@aic.nrl.navy.mil (202) 767-2684

Abstract

In domains where a stochastic process is involved in the evaluation of a candidate solution, multiple evaluations are necessary to obtain a good estimate of the performance of an individual. This work shows that biasing the sampling of that problem configuration space can lead to better performance of the structure being learned given the same amount of effort.

1 Introduction

In many domains, particularly those where a stochastic or noisy evaluation process is involved, the evaluation of a candidate solution might require sampling, i.e. multiple evaluations, to get a good estimate of performance. This random sampling over the space of possible configurations of the problem environment is typically performed with a uniform distribution. In some cases, better performance can be achieved by using a non-uniform distribution of samples from this problem configuration space. Furthermore, instead of randomly choosing samples with a fixed distribution, the distribution can be altered *adaptively* over time to achieve a particular goal. We call this **adaptive sampling** of the problem configuration space.

It is important to note that the problem configuration space to which we refer is *not* the same space as the solution search space. The **solution search space** is the space that the genetic algorithm searches, i.e. the space where crossover and mutation are applied. This is the space of candidate solutions. The **problem configuration space** is the space of possible variations in the form of the problem being solved. For example, in the Evasive Maneuvers (EM) domain,[1] several parameters define the characteristics of the missile we are evading and the initial starting conditions of the missile in relation to the plane. Each evaluation, from the Genetic Algorithm (GA) point of view, requires multiple episodes of simulation, where the parameters of the missiles and starting conditions are randomly chosen each time. The performance is the average over these episodes. The space of possible parameter settings defines the problem configuration space, and adaptive sampling biases the distribution of samples from this space. The bias[2] is adjusted adaptively based on the performance of the samples seen. In essence, we are adaptively altering the evaluation function to learn a better global solution over the problem configuration space.

Many complex domains require multiple evaluations of the candidate solutions to get a good estimate of their performance. For example, some domains involve a simulation of an environment where the initial configuration of the environment is randomly chosen each episode (Schultz, 1991; Grefenstette, 1991; Selfridge, Sutton and Barto, 1985). In other domains, the evaluation process itself is noisy due to computational constraints. For example, in Fitzpatrick and Grefenstette (1988), an image registration process used statistical sampling of the image to reduce the computational complexity of the evaluation. The resulting noisy evaluation eliminated the need to examine all pixels in an image.

These complex domains typically use a uniform distribution in randomly selecting each sample from the space of problem configurations, in order to gain an accurate estimate of candidate solutions. What we are interested in is having the learning system choose the samples from the configuration space non-uniformly to maximize some aspect of the learning, e.g., the average performance over the entire configuration space. The learner, in this case, might be analogous to an active learner in that it chooses the specific training environment to maximize its learning.

There are several motivations for wanting to alter the selection of samples. In a general sense, we want our learning system to acquire knowledge structures that perform as well as possible in the domain of interest. In particular, we want the learned knowledge structures to be as generally useful as possible, while retaining high performance. It is well-known that randomly sampling the space of initial configurations of a problem yields more robust solutions (Sammut and Cribb, 1990). However, if this space of configurations is very large with much irregularity, then it is difficult to adequately sample enough of the space. Adaptive sampling tries to include the most productive samples so that the amount of sampling of the problem configuration space is reduced. Even if the space is not too large and has enough regularities to sam-

[1] The EM domain (Erickson and Zytkow, 1988) will be described in more detail in the next section.

[2] Here, we define bias to mean the non-uniform distribution.

ple adequately, adaptive sampling allows selecting a criteria over which the samples are chosen. The learned knowledge structures can be adapted to a particular use by biasing the sampling of the problem configuration space. Given an effort equal to uniform sampling, higher performance knowledge structures can be produced. This last use of adaptive sampling will be demonstrated in this study.

While the GA is sampling the solution space for good solutions to the problem, the adaptive sampling mechanism is sampling the space of possible configurations of the problem to be solved, in an effort to maximize what is learned by the GA. But what do we mean by "maximizing what is learned by the GA?" Depending on the goal, we might want to learn something that works as well as possible over the entire problem configuration space, or we might want a solution that produces a uniform performance over the area, i.e. we might be willing to accept a lower mean performance if the same performance is achieved at all points in the space. Another goal might be to maximize some subarea of the space.

To measure the above goals, we will use various statistics, such as the mean of performance over the area, the variance of the performance, the maximum or minimum of the performance, or the area of the space that achieves some level of performance.

This paper will show that without increasing the effort required, a higher performance solution can be found by biasing the sampling of the problem configuration space. This will be demonstrated empirically with the addition of an adaptive sampling mechanism to SAMUEL, a GA-based learning system that learns strategies for solving sequential decision problems.

Actively selecting training examples is not a new concept. In Scott & Markovitch (1989), a conceptual clustering system used a heuristic to guide the search of experience space, such that informative training examples could be generated. The heuristic was based on Shannon's uncertainty function. Although uncertainty is a useful heuristic when trying to maximize what is known about a region of the search space, in this study we base our biasing on the performance of regions of the space. The reason for this is two-fold. First, we assume that performance is the only feedback available to us, and we want to reduce the amount of explicit bookkeeping we must perform. Second, as will be shown later, we want to affect our sampling in ways that are related to the performance of the system.

Section 2 will identify the problem configuration space with respect to the Evasive Maneuvers domain. Section 3 will describe the adaptive sampling mechanism used in these experiments for sampling the configuration space. The experimental methodology will be explained in Section 4, and the results presented in Section 5. Finally, Section 6 will summarize the experiments and suggest better mechanisms for performing the sampling.

2 The EM Domain and Problem Configuration Space

The **Evasive Maneuvers** (EM) domain is a two-dimensional missile and plane problem where the object is for the plane to avoid being hit by the missile. The missile is initially much faster than the plane, but the plane is slightly more maneuverable. The missile will eventually exhaust its fuel and fall from the sky. The use of SAMUEL to learn plans for this domain was presented in Grefenstette, Ramsey, and Schultz (1990). The empirical results in this paper use the EM domain.

In this domain, missiles can have a wide variety of characteristics. In particular, the two main attributes for missiles are their initial speed and their maneuverability, and we have a rough idea of the minimum and maximum values for these attributes. Therefore, we will define our problem configuration space as a two dimensional space where one dimension corresponds to the missile speed, and the other dimension corresponds to the missile maneuverability.

Although we can learn high-performance strategies for evading specific instances of missiles from this space, we also want to learn a single strategy that has relatively high performance over as much of the space as possible. The system aboard the plane might then use the specialized strategies to defend against specific, *known* missiles, but will also have a general default strategy to use between the time a missile is first detected, and the time the missile is classified as a particular type.

Adaptive sampling will allow us to generate a high performance general strategy over a larger area of the problem configuration space. With uniform sampling, it would not be possible to learn to perform as well on the entire space.

3 An Adaptive Sampling Mechanism

This section describes the adaptive sampling mechanism in SAMUEL. Please note, however, that the specific mechanism used here is only one possible instantiation. Other mechanisms are possible, and will be discussed in the conclusion.

Outside the GA, there is a two-dimensional matrix. One dimension of the matrix corresponds to the missile speed and the other corresponds to the missile maneuverability. Each cell of this matrix represents a *gross* estimate of the performance for that combination of missile speed and maneuverability *over all episodes, over all members of the population, and across all generations.* The cells are initialized with the average possible payoff. This estimate is updated *every* episode with the following calculation:

$$CellValue = CellValue + B(reward - CellValue)$$

where B represents a rate of learning, *reward* is the performance from the episode, and *CellValue* is the current value of the cell in the matrix being updated. *CellValue*

will converge to the mean payoff for the associated configuration.

The matrix is used *each* episode to bias the selection of the missile characteristics that will be used in that episode of the evaluation. Exactly how the matrix is used depends on the goal of using adaptive sampling. Many disciplines for biasing the distribution may be implemented. In this study, two biasing schemes are examined: **inverse bias**, and **contour bias**. In each case, the bias defines a weighting for the distribution of samples, and the samples are chosen stochastically based on the weighting.

The first criterion examined for biasing the distribution of samples, inverse bias,[3] can be stated as follows:

- Sample more heavily from areas of the space with lower performance, but never stop sampling the good areas.

This bias is shown in graphical form in Figure 1, where the X-axis is the value from the matrix (i.e., the input to the weighting function), and the Y-axis is the weighting for the selection of the sample.

FIGURE 1: Inverse Bias function.

The intention here is that heavier sampling of the worst performing areas will force the system to "concentrate" on improving those areas of the problem configuration space. We want to continue to sample from the areas that already perform well so that we do not *forget* what we know for that area. One drawback of this approach is the underlying assumption that additional training on poor performing regions in the configuration space will improve the performance in these regions, or that the boundaries of the configuration space can be selected such that only "learnable" areas are included. If the total area of the problem configuration space includes a large area that can not be learned because of limited capabilities of the learning agent, then the overall performance will degrade. In the EM domain, if the missile is fast enough and manueverable enough, then no amount of learning will allow the aircraft to escape the missile. Therefore, it is important to limit the entire configuration space to areas known to be learnable. However, this might not be possible in practice. This observation was the motivation for the next criterion.

The next criterion used for biasing the distribution, contour bias,[4] can be stated as:

- Define a parabola shaped weighting around some performance value so that you sample more in areas that are close to that performance level, but never stop sampling at a fair rate in areas above that performance, and sample a little in areas below that performance.

This weighting function is illustrated in Figure 2.

FIGURE 2: Contour Bias function.

the contour bias technique tends to *push* the area of at least the chosen performance out to cover a greater region, but does not suffer the problem of trying to sample areas where there is no hope of achieving any improvement. An improvement over the last criterion is that non-learnable regions are avoided. Whereas the last criterion tends to improve the mean performance over the entire area, this method is good for expanding the region of some given level of performance. This level of performance must be specified, and in Figure 2, as well as the reported experiments, is set at 90 percent.

Other disciplines are possible, depending on the objective of biasing the sampling. The results of applying these two adaptive sampling techniques is presented next, along with the results from the baseline (uniform distribution) experiment.

[3] So called because the distribution weighting is the inverse of the performance.

[4] This name refers to the distribution following a "contour" of a given performance level.

4 Experimental Method

In order to test the effectiveness of adaptive sampling, the following methodology was used. Each experimental run is composed of the learning stage, where an optimal plan is learned, followed by a testing stage, where that plan is evaluated.

TABLE 1: Statistics for uniform and adaptive sampling.

	uniform	inverse	contour
mean	86.86	91.17	92.13
variance	189.21	119.01	139.70
minimum	20.29	25.29	24.90
maximum	99.40	99.59	100.00
area above 95%	33%	51%	65%
area above 98%	13%	16%	35%
area above 99%	2%	2%	18%

During the learning phase, the adaptive sampling mechanism is enabled using one of the biases for choosing the samples from the problem configuration space. After 100 generations, the best plan is retrieved. In the testing phase, this best plan is subjected to extended evaluations on 256 combinations of values for the two parameters that define the problem configuration space (i.e. each parameter is divided into 16 values). The performance is presented as a contour plot where the x-axis is the missile speed and the y-axis is the maximum missile turning rate (maneuverability). The contours represent the level of performance with a given combination of speed and turning rate.

For comparative purposes, a baseline experiment was performed without adaptive sampling. In this experiment, random sampling with uniform distribution was performed over the problem configuration space. This would be equivalent to performing a Monte Carlo sampling of the space. The best plan from this experiment was again tested over the combination of values for the two parameters as described above.

In addition to the contour plots, which give a visual picture of the performance of the plan over the problem configuration space, various statistics can quantify the overall effect, as discussed earlier. For each experiment, we measure the mean performance over the space, the variance of the performance, the maximum and minimum performance, and the percentage of the area of the space where the performance was greater than or equal to 95, 98 and 99 percent. Table 1 summarizes the results for the baseline case of uniform distribution and for the two non-uniform distributions, inverse bias and contour bias.

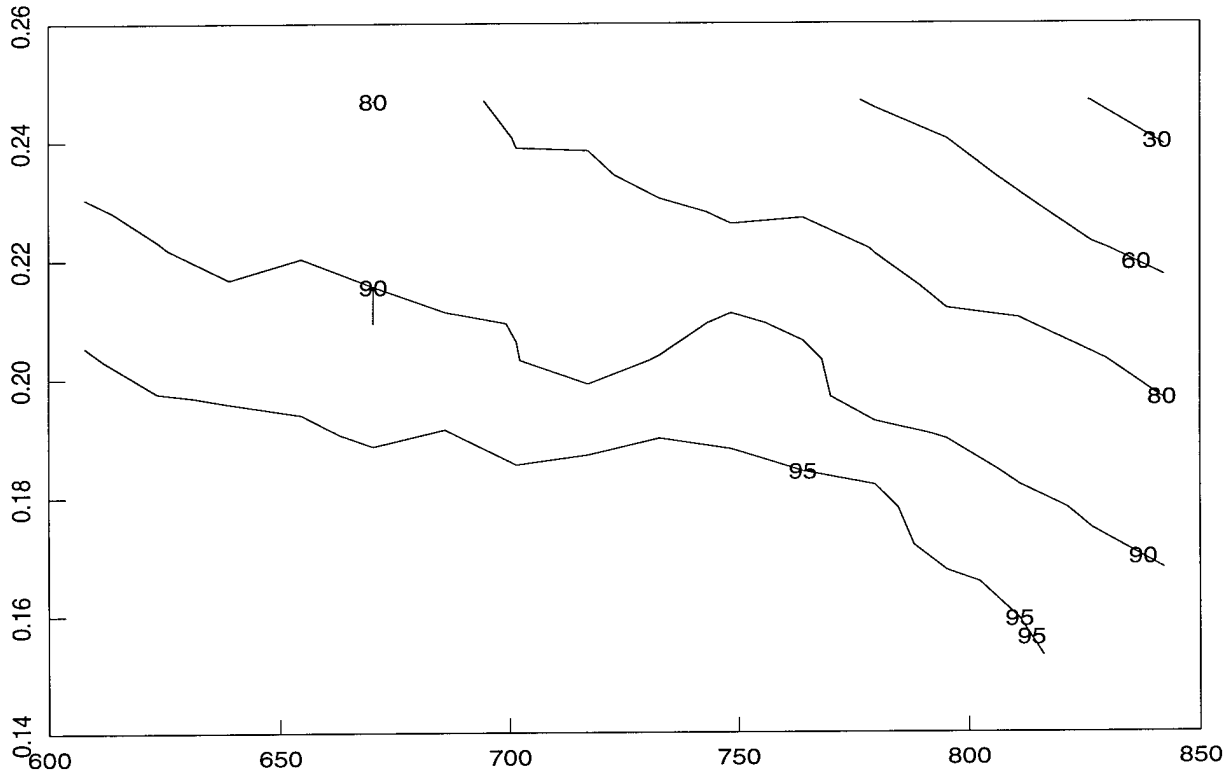

FIGURE 3: Baseline performance using uniform distribution.

5 Results

The results of the baseline experiment are shown in Table 1, under the heading *uniform*, and in Figure 3. Here we can see the effect of training over the entire problem configuration space using a uniform distribution of samples during the training. The results for the first adaptive sampling discipline, inverse bias, are shown in Table 1, under the heading *inverse*, and in Figure 4. These show the effect of sampling more where the performance of the samples is lower. The results from the second adaptive sampling discipline, contour bias, are shown in Table 1, under the heading *contour*, and in Figure 5. The results of biasing the samples towards a particular performance to expand its region is shown.

By comparing the columns in Table 1 and examining the three figures, the advantage of biasing the sampling becomes clear. The uniform sampling, as seen in Figure 3, only has acceptable performance on a small portion of the space. The inverse bias, by definition, strives to get a uniformly good performance over the entire area by concentrating on those areas where performance is lower. We can see from Figure 4 that the area of good performance is much larger. From the statistics, we see that the mean performance is better than in the uniform sampling case, and in particular, the variance in performance over the area is much lower. Also, the minimum performance in the area has risen significantly.

This indicates that the strategy learned is more robust and generally applicable to more of the situations it might encounter within the problem configuration space. Notice, however, that the performance at 98 percent and above did not increase significantly. Suppose that instead of wanting a strategy that performed relatively well, where the emphasis was on uniformity over the space, we wanted to emphasize a higher performance strategy, and find out how much of the space we could get it to cover.

With the contour bias, the emphasis is on making the area of very high performance as large as possible. This is achieved by concentrating on the area around a given, relatively high performance. This area should then "grow" to cover more area. In the table, we can see that the areas of very high performance are much greater when using the contour bias. In particular, the area with a performance greater than or equal to 98 percent has nearly tripled, while the area above 99% has increased by almost a factor of ten. A side effect, however, is that the variance is higher than in the inverse bias case. This is to be expected, since inverse bias strives for uniformity, while the contour bias attempts to enlarge the area of good performance, sometimes at the expense of other areas of the space. Note, however, that the variance is still better than when not using adaptive sampling at all. This bias also gave the highest maximum performance.

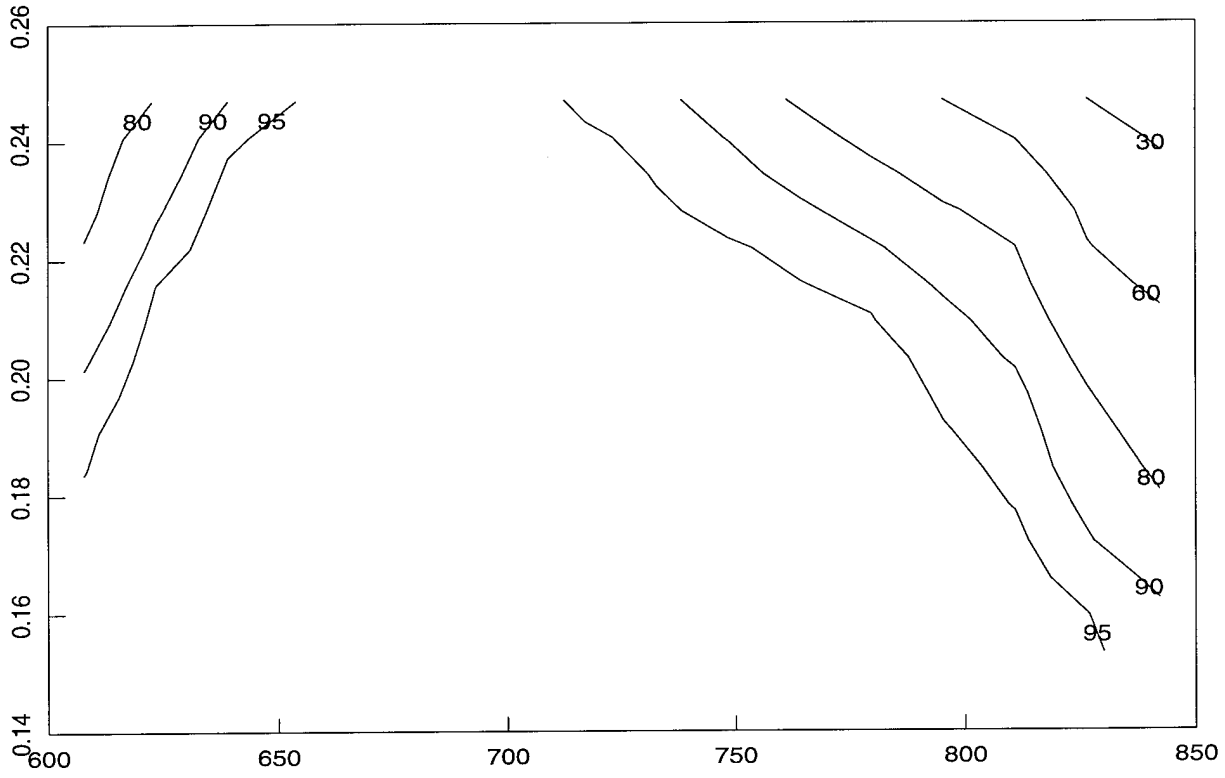

FIGURE 4: Performance using inverse bias.

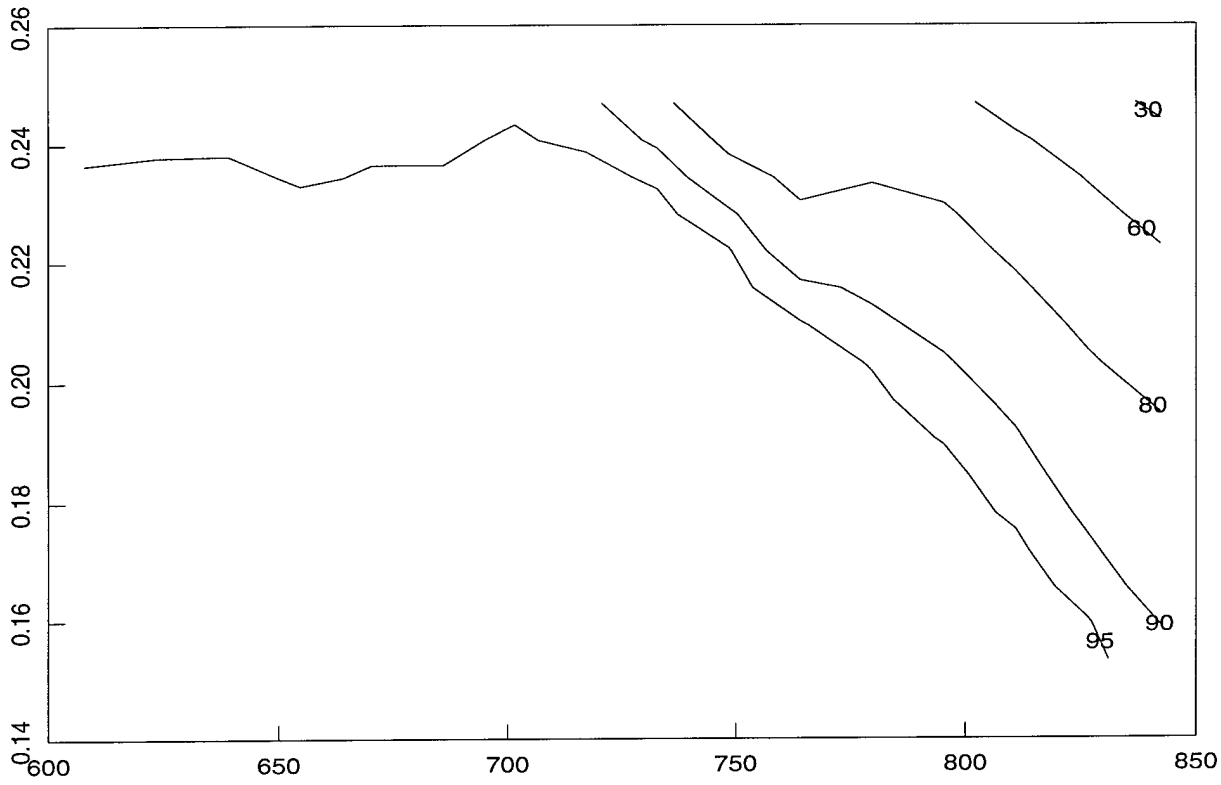

FIGURE 5: Performance using contour bias.

6 Conclusion

The empirical evidence presented in this paper shows that biasing the sampling of the problem configuration space can improve the overall performance (along some dimension) of the knowledge structures learned in problems involving stochastic evaluations. In the EM domain, the adaptive sampling is successfully used to generate a strategies with specific properties that are useful for a wide variety of opponents, by biasing the sampling of the space of opponents.

Future research will examine adaptive sampling in other domains, including domains where the motivation of reduced effort in sampling can be tested. Also, we would like to examine and characterize other biasing disciplines.

This paper's intention is to demonstrate the *idea* of adaptive sampling. The method used here (a two dimensional matrix) will not scale up if the problem configuration space is more than a few dimensions. In fact, the natural choice for the mechanism for adaptive sampling is a genetic algorithm, since it embodies the notion of implicit statistics without explicit bookkeeping. Therefore, another direction for further research is to develop adaptive sampling as a meta or cooperative genetic algorithm.

Acknowledgements

The author wishes to thank Helen Cobb and Connie Ramsey for their useful comments on the paper, the referees for their careful reviews, and the machine learning group at NCARAI for their comments on the idea of adaptive sampling, particularly John Grefenstette. This work is supported in part by ONR under Work Request N00014-91-WX24011.

References

Erickson, M. D. and J. M. Zytkow (1988). Utilizing experience for improving the tactical manager. *Proceedings of the Fifth International Conference on Machine Learning.* Ann Arbor, MI. (pp. 444-450).

Fitzpatrick, J. M. and J. J. Grefenstette (1988). Genetic algorithms in noisy environments. *Machine Learning, 3(2/3),* (pp. 101-120).

Grefenstette, J. J. (1991). A lamarkian approach to learning in adversarial environments. *Proceedings of the Fourth International Conference on Genetic Algorithms.* San Diego, CA: Morgan Kaufmann.

Grefenstette, J. J., C. L. Ramsey, and A. C. Schultz (1990). Learning sequential decision rules using

simulation models and competition. *Machine Learning, 5(4),* (pp. 355-381).

Grefenstette, J. J. and J. M. Fitzpatrick (1985). Genetic search with approximate function evaluations. *Proceedings of the First International Conference on Genetic Algorithms and Their Applications.* Pittsburgh, PA: Lawrence Erlbaum Assoc. (pp.112-120).

Sammut, C. and J. Cribb (1990). Is learning rate a good performance criterion for learning? *Proceedings of the Seventh International Conference on Machine Learning.* Austin, TX: Morgan Kaufmann. (pp. 170-178).

Scott, P., and S. Markovitch (1989). Learning novel domains through curiosity and conjecture. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence,* Detroit, MI, pp. 669-674

Schultz, A. C. (1991). Using a genetic algorithm to learn strategies for collision avoidance and local navigation. *Proceedings of the Seventh International Symposium on Unmanned Untethered Submersible Technology.* Durham, NH: IEEE.

Selfridge, O. G., R. S. Sutton and A. G. Barto (1985). Training and tracking in robotics. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence.* Los Angeles, CA: Morgan Kaufmann. August, 1985. (pp 670-672).